Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD

naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

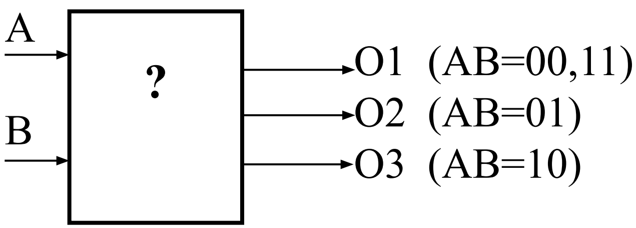
Q17

In an algorithm implemented in hardware, two bits (say A and B) are checked at each step to determine one of the three operations:

- Operation 1, if (A=0, B=0) or (A=1, B=1)
- Operation 2, if (A=0, B=1)
- Operation 3, if (A=1, B=0)

Design the required logic circuit.

A17

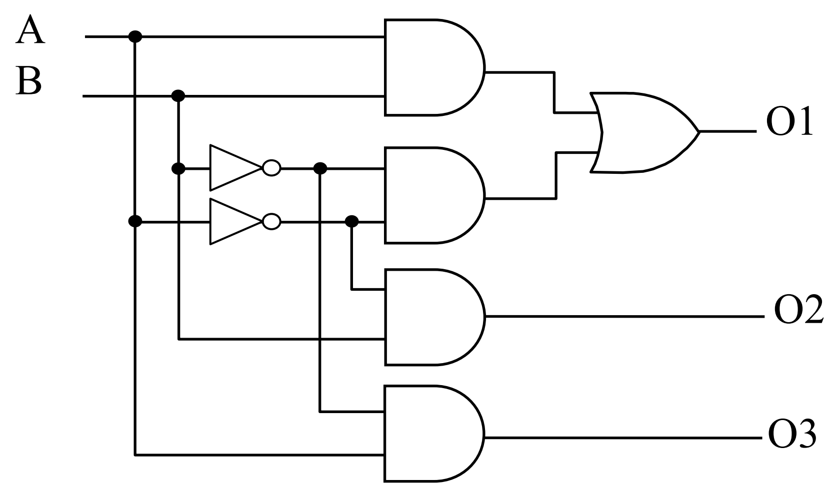


A	B	O1	O2	O3
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

$$O1 = \overline{A}\,\overline{B} + A\,B$$

$$O2 = \overline{A}\,B$$

$$O3 = A\,\overline{B}$$



Q18

X is a decimal number and its value is 51.

(a) Represent X in unsigned binary integer form using 8 bits,

(b) Find the corresponding hexadecimal representation of X,

(c) Assuming X is a 2s complement signed integer number using 8 bits, find –X.

(d) Represent X in BCD

A18

(a) $X = (51)_{10} = (00110011)_2$

(b) $X = (51)_{10} = (0011\ 0011)_2 = (33)_{16}$

(c) $X = (51)_{10} = (00110011)_2$ □ $-X = -(51)_{10}$
$= 2\text{s complement of } (X)_2 = (11001101)_2$

(d) $X = (51)_{10} = (0101\ 0001)_{BCD}$

Q19

The following table shows a memory in a computer system.

(a) How many memory locations this memory has?

(b) What is the memory size (in Bytes)?

(c) Total how many bits this memory can store?

Memory location	Memory Content
000000	1010110010101100
000001	0010110110101100
.	.
.	.
.	.
111110	1010110010111101
111111	0000110110101100

A19

(a) Because the address field of this memory is 6 bits, there are $2^6 = 64$ memory locations.

(b) Each location stores 2 Byte (16 Bits) data. So the memory size is number of locations × size of data at each location = $64 \times 2 = 128$ Byte.

(c) Because a byte is 8 bits, this memory can store maximum $128 \times 8 = 1024$ bits data.

Q20

If one line of an assembler program for a hypothetical processor, which uses bytewise addressing, and corresponding machine code is given as:

address	**assembler**	**machine code**
2000	add F1FA, B1B1	12F1FAB1B1

(a) How many instructions this processor might have?

(b) How many bits are needed for the instruction register?

(c) How many memory location an instruction will occupy?

(d) What is the total directly addressable memory size ?

A20

address	assembler	machine code
2000	add F1FA, B1B1	12F1FAB1B1

(a) Because the op-code field of this instruction is 2 hexadecimal digits (8 bits) there might be maximum $2^8 = 256$ instructions

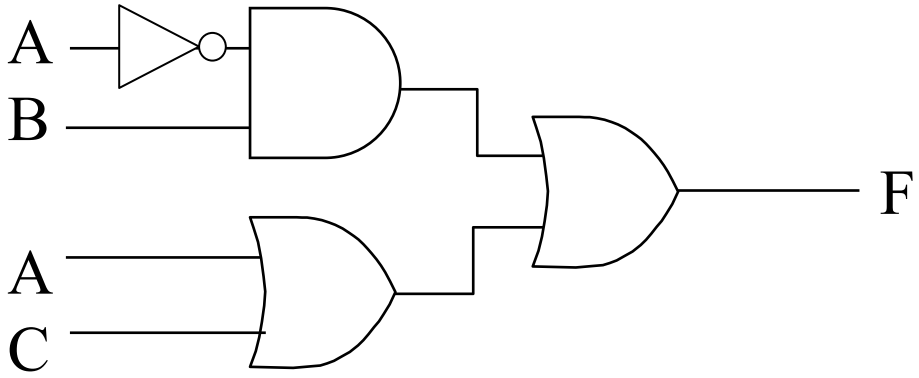
(b) Because the instruction length is 10 hexadecimal digits, at least $4\times10 = 40$ bits are needed for the instruction register.

(c) Each location stores 1 Byte (8 Bits) data. The instruction length is 5 bytes. Therefore each instruction will occupy 5 memory locations.

(d) Address field of this instruction has 4 hexadecimal digits, so the total memory size is $2^{4\times4} = 2^{16}$ Byte $= 2^{10+6}$ Byte $= 2^6 \times 2^{10}$ Byte = 64 kByte.

Q21

Given the following network;



(a) Write an expression for the network,

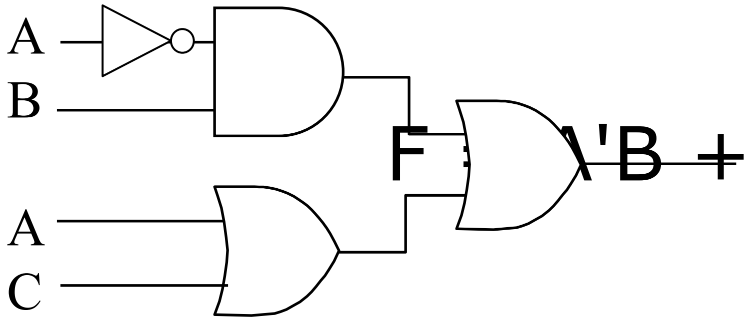
(b) Simplify the expression.

(c) Write out a truth table for the expression.

(d) Draw the network of the simplified expression.

A21

(a)



F = A'B + A + C

(c) Truth table:

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

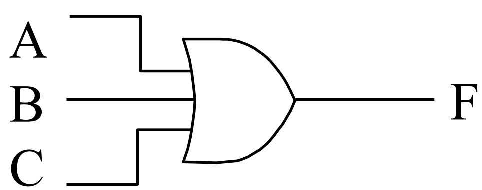
(b)

$F = A'B + A + C$

$F = (A + A')(A + B) + C$

$F = A + B + C$

(d)



Q22

Convert the given decimal number $(-18.75)_{10}$ to the corresponding 32 bit IEEE floating point number.

A22

- Step 1. Convert the number to binary

  $(18.75)_{10} = (10010.11)_2$

- Step 2. Normalize the binary number to 1

  $(10010.11)_2 = (1.001011 \times 2^4) = (1.001011 \times 2^{00000100})$

- Step 3. Add bias (127) to exponent to obtain biased exponential format

  biased exponential format = $(1.001011 \times 2^{4+127}) = (1.001011 \times 2^{00000100+01111111}) = (1.001011 \times 2^{10000011})$

  S = 1, BE = 10000011, M = 00101100000000000000000

- Step 4. Join S, BE and M to form binary FP number

  $(-18.75)_{10} = (1\ 10000011\ 00101100000000000000000)_{fp}$

  $(-18.75)_{10} = (C1960000)_{\text{fp in hexadecimal}}$

Q23

Write a simple VVM Assembly program that adds *the number entered from the keyboard* and *the data in memory location **20***.

If the result is positive, it is saved in memory location **30**;

If the result is negative, it is saved in memory location **40**

A23

One solution is as follows:

```
00 in        // input number
01 add 20    // accumulator + content of (20)
02 brp 05    // if +ve jump to 05
03 sto 40    // else store –ve number in (30)
04 jmp 06    // jump (branch) to 06
05 sto 30    // save the +ve number in (30)
06 hlt       // stop
```

Q24

- Consider an array of five drives (X0, X1, X2, X3 contain data, X4 is parity disk).
- Parity of *i*th bit is calculated as $X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i)$
- Suppose that drive **X2** has failed.
- Show how to regenerate the contents of **X2.** ?

A24

- The contents of X2 can be regenerated by XORing both sides by X4 and X2 :

$$X2(i) \oplus X4(i) \oplus X4(i) = X3(i) \oplus X2(i) \oplus X1(i) \oplus X0(i) \oplus X2(i) \oplus X4(i)$$

- Because $X4(i) \oplus X4(i) = 0$ and $X2(i) \oplus X2(i) = 0$;

$$X2(i) = X4(i) \oplus X3(i) \oplus X1(i) \oplus X0(i)$$